



King's Research Portal

Document Version

Version created as part of publication process; publisher's layout; not normally made publicly available

[Link to publication record in King's Research Portal](#)

Citation for published version (APA):

Angelini, O., & Di Matteo, T. (Accepted/In press). Complexity of products: the effect of data regularisation. *Entropy*.

Citing this paper

Please note that where the full-text provided on King's Research Portal is the Author Accepted Manuscript or Post-Print version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version for pagination, volume/issue, and date of publication details. And where the final published version is provided on the Research Portal, if citing you are again advised to check the publisher's website for any subsequent corrections.

General rights

Copyright and moral rights for the publications made accessible in the Research Portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognize and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the Research Portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the Research Portal

Take down policy

If you believe that this document breaches copyright please contact librarypure@kcl.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

Article

Complexity of products: the effect of data regularisation

Orazio Angelini ^{1,†,‡,*} and T. Di Matteo ^{1,2,3,†,‡}

¹ Department of Mathematics, King's College London, The Strand, London, WC2R 2LS, UK

² Department of Computer Science, University College London, Gower Street, London, WC1E 6BT, UK

³ Complexity Science Hub Vienna, Josefstädter Strasse 39, A 1080 Vienna, Austria

* Correspondence: orazio.angelini@kcl.ac.uk

† Current address: Department of Mathematics, King's College London, The Strand, London, WC2R 2LS, UK

‡ These authors contributed equally to this work.

Academic Editor: name

Version October 12, 2018 submitted to Entropy; Typeset by L^AT_EX using class file mdpi.cls

Abstract: Among several developments, the field of Economic Complexity (EC) has notably seen the introduction of two new techniques. One is the Bootstrapped Selective Predictability Scheme (SPSb), which can provide quantitative forecasts of the Gross Domestic Product of countries. The other, Hidden Markov Model (HMM) regularisation, denoises the datasets typically employed in the literature. We contribute to EC along three different directions. First, we prove the convergence of the SPSb algorithm to a well-known statistical learning technique known as Nadaraya-Watson Kernel regression. The latter has significantly lower time complexity, produces deterministic results, and it is interchangeable with SPSb for the purpose of making predictions. Second, we study the effects of HMM regularization on the Product Complexity and logPRODY metrics, for which a model of time evolution has been recently proposed. We find confirmation for the original interpretation of the logPRODY model as describing the change in the global market structure of products with new insights allowing a new interpretation of the Complexity measure, for which we propose a modification. Third, we explore new effects of regularisation on the data. We find that it reduces noise, and observe for the first time that it increases nestedness in the export network adjacency matrix.

Keywords: Complex Systems; Economic complexity; Fitness; Complexity; Regression; Nestedness; Hidden Markov Model; Regularization

Contents

1	Introduction	2
2	Results	3
2.1	Convergence of SPSb to a Nadaraya-Watson kernel regression	3
2.1.1	Analytical convergence	5
2.1.2	Numerical convergence	7
2.2	HMM regularization reduces noise and increases nestedness	7
2.3	Predictions on products with SPSb	10
2.4	Figures, Tables and Schemes	12

27	3 Materials and Methods	18
28	3.1 Fitness and Complexity algorithm	18
29	3.2 LogPRODY	19
30	3.3 Complexity-logPRODY motion model	20
31	3.4 SPSb	20
32	3.5 Nadaraya-Watson kernel regression	21
33	3.6 HMM regularization	22
34	3.7 Datasets and product digits	22
35	4 Conclusions	23
36	A Country predictions via the products	24

37 1. Introduction

38 Complexity and Fitness measures were originally proposed [1] within the field of Economic
 39 Complexity (EC) to capture respectively the level of sophistication of a given class of products
 40 found on the international export market and the advancement of the productive system of a
 41 country. These two measures are calculated from international trade data, and they stem from
 42 the hypothesis that the difference between countries' competitiveness comes from their respective
 43 *capabilities* [2–4]. Capabilities are non-exportable features of the productive system of a country
 44 that allow it to produce a certain class of products. The problem with the theory of capabilities is
 45 that capabilities themselves are hard to define: one can speculate on what they might be, e.g. good
 46 regulations, a well-organized education system, or maybe the presence of facilities specifically useful
 47 for a product's making, but there is currently no good principled “*a priori*” or normative approach to
 48 classify and measure them [5]. On the other hand, the observation that a country c exports product p
 49 contains a strong signal. It implies that c is competitive enough in the production of p for export to
 50 be convenient on the global market. Therefore, one could say that c has all the capabilities needed
 51 to make p . Hausmann [6] proposed the *Method of Reflections*, a non-normative algorithm to rank
 52 countries by how many capabilities they have, and products by how many capabilities they need for
 53 production, based on observed exports. The algorithm leverages topological properties of the export
 54 network, which is a bipartite network where the nodes can be either countries or product classes,
 55 and where a link is added to the network if country c is a significant exporter of p . Fitness and
 56 Complexity are the output of an alternative algorithm [7] exploiting the discovery that the export
 57 network has a nested topology [1] (a comparative analysis is found in [8]). In other words, it has
 58 been observed that some countries, usually the richest in monetary terms, export almost all product
 59 classes, and some products are exported only by the countries that are most diversified in terms of
 60 export. Conversely, the less diversified countries only export a handful of products which are also
 61 being exported by almost all countries. This means that the adjacency matrix of the export network
 62 M_{cp} can be reordered to be very close to triangular, in analogy with some biological systems [9,10].
 63 The Fitness/Complexity algorithm takes the adjacency matrix M_{cp} as an input and produces a value
 64 of Fitness F for each country and one of Complexity C for each product. Sorting the matrix rows and
 65 columns by increasing Fitness and Complexity produces the characteristic triangular structure. This
 66 ordering offers a robust way to rank the countries in terms of their competitiveness and products in
 67 terms of how sophisticated they are [1]. Nestedness of the bipartite export network is a fundamental
 68 point of the theory and, in this paper, we measured nestedness with one widespread metric, NODF
 69 [11], for the first time. The Economic Complexity approach is an innovative way to use the wealth of
 70 data that is being currently produced in economics, and it has the advantage of offering a data-driven
 71 and mathematically defined method of analysis, which reduces the necessity of interpretation.

72
 73 Several results have been produced in many directions but mainly in the direction of the Fitness
 74 measure. The network approach produced an algorithm to forecast the sequence of products a

country will start to export [12], and inspired the exploration of innovation models [13]. Fitness as a macroeconomic indicator has been particularly fruitful. One very interesting result calls for an extension of neo-classical economic theories of growth. It is classically understood that for countries to start the process towards industrialization they have to pass a threshold of GDP per capita (GDPpc), and it has been found that higher Fitness can significantly lower this threshold [14]. It has long been observed that Fitness might allow for Gross Domestic Product (GDP) prediction [1,15], but the most recent advances have introduced a dynamical systems based approach to quantitative forecasting called *Bootstrapped Selective Predictability Scheme* [16] (SPSb, see Section 3.4). The method is based on the observation that trajectories of countries tend to be collinear in many regions in the GDP-Fitness two-dimensional space. Making the assumption that the growth process of countries can be modelled as a two-dimensional dynamical system allows to use nonparametric regression techniques such as the *method of analogues* [17] to forecast growth. SPSb been proven to give state-of-the-art GDP forecasts [18]. In this work, we prove that SPSb converges to a well-known nonparametric regression originally proposed by Nadaraya and Watson. The same work introduced a new regularization method for the M_{cp} based on a Hidden Markov model (HMM, see Section 3.6), and it has been proven to give state-of-the-art GDP forecasts [18] (but, to our best knowledge, has never been applied to the Complexity measure until the present work). These ideas were originally introduced to validate the new Fitness metric, which is non-monetary, by comparing and contrasting it to an established monetary metric such as GDP. This line of thinking proved very fruitful, so other attempts have been made to extract information by comparing an Economic Complexity metric with established ones. One such attempt compared economic inequality measurements with Fitness [19]. This paper contributes to the latest developments of the Complexity and Fitness measures and it follows up mainly from the earlier work by Angelini et al. [20] focusing on the Complexity measure. In particular, the Complexity index has been paired with *logPRODY* (L , see Section 3.2) to obtain an interesting insight. *LogPRODY* of a product is a weighted average of the GDP of its exporters, where the weights are proportional to comparative advantage in making that product. It is possible to represent product classes as points on the Complexity-*logPRODY* plane. Their motion on said plane can be modelled with a potential-like equation [20] (see Section 3.3 for more details). In this work, we report the results of the application of SPSb and HMM regularization on the Complexity measure, and we show how HMM affects the M_{cp} matrices.

This paper is structured as follows. In Section 2.1 we show that, as suggested in [16], the SPSb technique converges to the faster and mathematically well-grounded Nadaraya-Watson kernel regression (NWKR), allowing applications of SPSb to larger datasets. In Section 2.2 we look at how the HMM regularization affects the aforementioned Complexity-*logPRODY* plane motion and analyse its effect on a set of different M_{cp} matrices. Finally, Section 2.3 reports our application of the SPSb algorithm to make predictions on the Complexity-*logPRODY* plane.

2. Results

2.1. Convergence of SPSb to a Nadaraya-Watson kernel regression

In this section, we prove that the SPSb prediction method converges, for a large number of iterations, to a Nadaraya-Watson kernel regression (NWKR). The idea was originally suggested in [16], but never developed mathematically. We prove the convergence analytically and numerically so that for all prediction purposes the two methods are interchangeable. The result is significant because it connects SPSb to a well-established, tried and tested technique, and frames the predictions made with this method in a more mathematically rigorous setting. SPSb is a non-deterministic algorithm so, at every run, it will yield slightly different results, while NWKR will always produce the same results up to machine precision. From a computational perspective, NWKR has much smaller time

complexity, so our result allows the use of SPSb on much larger datasets than previously explored.

SPSb is fundamentally a nonparametric regression. We describe the algorithm here, and in Section 3.4. In the original formulation [18], one is presented with $\vec{x}_{\hat{c},\hat{t}}$, the position of a given country \hat{c} in the Fitness-GDP (FG) plane at time \hat{t} , and wants to predict the change (displacement) in GDP at the next timestep $\hat{t} + \Delta t$, namely $\delta x_{\hat{c},\hat{t}}$. The method is based on the idea, advanced in [15], that the growth process of countries is well modeled by a low-dimensional dynamical systems. For many important cases, the best model is argued to be embedded in the two-dimensional Euclidean space given by Fitness and GDPpc. It is not possible to identify the analytical equations of motion, so instead one uses observations of previous positions and displacements of other countries $(\delta x_{c,t}, \vec{x}_{c,t})$, which are called *analogues*, a term borrowed from [17]. Because the evolution is argued to be dependent only on two parameters, observed past evolutions of countries nearby $\vec{x}_{\hat{c},\hat{t}}$ in the FG plane are deemed to be good predictors of $\delta x_{\hat{c},\hat{t}}$. Therefore SPSb predicts $\delta x_{\hat{c},\hat{t}}$ as a weighted average of past observations. The weights will be proportional to the similarity of country \hat{c} to its analogues, and the similarity is evaluated by calculating Euclidean distance on the Fitness-GDP plane. A close relative of this approach is the well-known K-nearest neighbours regression [21]. In order to obtain this weighted average, one samples with repetition a number B of bootstraps from all N available analogues. The sample probability density of an analogue $\delta x_{c,t}$, found at position $\vec{x}_{c,t}$ is given by a gaussian distribution:

$$p(\delta x_{c,t} | \vec{x}_{c,t}) = \mathcal{N}(\vec{x}_{\hat{c},\hat{t}} - \vec{x}_{c,t} | 0, \sigma), \quad (1)$$

$$\mathcal{N}(\vec{z} | \vec{\mu}, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{(\vec{z} - \vec{\mu})^2}{2\sigma^2}\right). \quad (2)$$

Therefore sampling probability will be inversely proportional to distance, i.e. analogues closer on the FG plane are sampled more often. We will adopt the following notation: each bootstrap will be numbered with b and each sampled analogue in a bootstrap with n , so each specific analogue sampled during the prediction of $\delta x_{\hat{c},\hat{t}}$ can be indexed with $s_{b,n}^{\hat{c},\hat{t}}$. Once the sampling operation is done, one averages the samples per bootstrap, obtaining $v_b^{\hat{c},\hat{t}} = \sum_n s_{b,n}^{\hat{c},\hat{t}} / N = \langle s_{b,n}^{\hat{c},\hat{t}} \rangle_n$. These averaged values constitute the distribution we expect for $\delta x_{\hat{c},\hat{t}}$. From this distribution we can derive an expectation value and a standard deviation (interpreted as expected prediction error) for $\delta x_{\hat{c},\hat{t}}$:

$$E_{\text{SPSb}}(\delta x_{\hat{c},\hat{t}}) = \frac{1}{B} \sum_{b=1}^B v_b^{\hat{c},\hat{t}}, \quad (3)$$

$$\sigma_{\text{SPSb}}^2(\delta x_{\hat{c},\hat{t}}) = \frac{1}{B-1} \sum_{b=1}^B \left(v_b^{\hat{c},\hat{t}} - E_{\text{SPSb}}(\delta x_{\hat{c},\hat{t}}) \right)^2 \quad (4)$$

$$. \quad (5)$$

Because closer analogues are sampled more, they will have a bigger weight in the averaging operations needed to compute expected value and standard deviation. The technique can be easily extended to other types of prediction, as we did in Section 2.3.

NWKR is conceptually very similar to SPSb. We will use the symbol \leftrightarrow to establish a correspondence between the two algorithms: in NWKR one is presented with an observation $X \leftrightarrow \vec{x}_{\hat{c},\hat{t}}$ and wants to predict $Y \leftrightarrow \delta x_{\hat{c},\hat{t}}$ from it. Other observations are available $(Y_i, X_i) \leftrightarrow (\delta x_{c,t}, \vec{x}_{c,t})$, and the prediction is a weighted average of the Y_i 's.

$$E(Y|X) = \frac{\sum_i K_h(X - X_i) Y_i}{\sum_i K_h(X - X_i)} \quad (6)$$

The weights will be given by K , a function of the distance on the Euclidean space containing the X_i values. This function is called *kernel*. A more detailed explanation of this technique can be found in Section 3.5.

2.1.1. Analytical convergence

SPSb returns both an expected value and a standard deviation for the quantity being measured. We begin by proving convergence of expected value.

Expected values. - Suppose that we execute B bootstraps of N samples from all available analogues $\{\delta x_{c,t}\}$, so that each sampled value in a bootstrap can be labelled as $s_{b,n}^{\hat{c},\hat{t}}$ with $1 \leq n \leq N$ and $1 \leq b \leq B$. Then the SPSb probabilistic forecast $E_{\text{SPSb}}(\delta x_{\hat{c},\hat{t}})$ will be:

$$E_{\text{SPSb}}(\delta x_{\hat{c},\hat{t}}) = \frac{1}{B} \sum_{b=1}^B \left(\frac{1}{N} \sum_{n=1}^N s_{b,n}^{\hat{c},\hat{t}} \right) = \frac{1}{BN} \sum_{b=1}^B \sum_{n=1}^N s_{b,n}^{\hat{c},\hat{t}}. \quad (7)$$

If we aggregate all B bootstraps, we can label the frequency with which the analogue $\delta x_{c,t}$ appears overall in the sampled analogues as

$$\phi_B^{\hat{c},\hat{t}}(\delta x_{c,t}) = \frac{1}{BN} \sum_{b=1}^B \sum_{n=1}^N \mathbf{1}_{\{\delta x_{c,t} = s_{b,n}^{\hat{c},\hat{t}}\}} \quad (8)$$

where $\mathbf{1}_{\{\cdot\}}$ is intended to be an indicator function. So we can rewrite the forecast as:

$$E_{\text{SPSb}}(\delta x_{\hat{c},\hat{t}}) = \sum_{c,t} \phi_B^{\hat{c},\hat{t}}(\delta x_{c,t}) \delta x_{c,t}, \quad (9)$$

where $\sum_{c,t}$ indicates a sum over all available analogues. But since the analogues are being sampled according to a known probability distribution $p(\delta x_{c,t} | \vec{x}_{\hat{c},\hat{t}})$, we can expect, by the law of large numbers, that for $B \rightarrow \infty$ the sample frequency will converge to the probability values (which it does, see Fig.2(a)):

$$\phi_B^{\hat{c},\hat{t}}(\delta x_{c,t}) \xrightarrow{B \rightarrow \infty} p(\delta x_{c,t} | \vec{x}_{\hat{c},\hat{t}}) \quad (10)$$

Now, SPSb uses a Gaussian probability distribution $p(\delta x_{c,t} | \vec{x}_{\hat{c},\hat{t}}) = \mathcal{N}(\vec{x}_{c,t} - \vec{x}_{\hat{c},\hat{t}} | 0, \sigma)$ (see Section 3.4) so our forecast will tend to:

$$E_{\text{SPSb}}(\delta x_{\hat{c},\hat{t}}) \xrightarrow{B \rightarrow \infty} \sum_{c,t} p(\delta x_{c,t} | \vec{x}_{\hat{c},\hat{t}}) \delta x_{c,t} = \sum_{c,t} \mathcal{N}(\vec{x}_{c,t} - \vec{x}_{\hat{c},\hat{t}} | 0, \sigma) \delta x_{c,t} \equiv E_{\text{NWKR}}(\delta x_{\hat{c},\hat{t}}), \quad (11)$$

but this is exactly the definition of a NWKR with Gaussian¹ kernel that has bandwidth σ (see Section 3.5). We assumed for brevity that the sum is already normalized, i.e. $\sum_{c,t} p(\delta x_{c,t} | \vec{x}_{\hat{c},\hat{t}}) = \sum_{c,t} \mathcal{N}(\vec{x}_{c,t} - \vec{x}_{\hat{c},\hat{t}} | 0, \sigma) = 1$, normalization is needed in Eqns.10,11 if this is not true, but it doesn't change the result of the proof.

¹ Note that in the machine learning literature it's usually not called Gaussian, but *radial basis function*.

Variances. - The variance of the distribution of samples in SPSb is calculated first by computing $v_b^{\hat{c},\hat{t}} = \sum_n^N s_{b,n}^{\hat{c},\hat{t}} / N = \langle s_{b,n}^{\hat{c},\hat{t}} \rangle_n$ i.e. the average of the samples of each bootstrap, and then computing the variance of the $v_b^{\hat{c},\hat{t}}$ across bootstraps, so (with the same notation as Eq.7) it can be written as:

$$\begin{aligned}
 \sigma_{\text{SPSb}}^2 &= \frac{1}{B-1} \sum_{b=1}^B \left(\frac{1}{N} \sum_n^N s_{b,n}^{\hat{c},\hat{t}} - \frac{1}{BN} \sum_{b',n'}^{B,N} s_{b',n'}^{\hat{c},\hat{t}} \right)^2 \\
 &= \frac{1}{B-1} \sum_{b=1}^B \left(v_b^{\hat{c},\hat{t}} - E_{\text{SPSb}}(\delta x_{\hat{c},\hat{t}}) \right)^2 \\
 &\approx \frac{1}{N} \sigma_{bn}^2(s_{b,n}^{\hat{c},\hat{t}}) \\
 &\equiv \frac{1}{N} \left(\frac{1}{(BN-1)} \sum_b^B \sum_n^N (s_{b,n}^{\hat{c},\hat{t}} - E_{\text{SPSb}}(\delta x_{\hat{c},\hat{t}}))^2 \right) \\
 &\approx \frac{1}{N} \left(\sum_b^B \sum_n^N \frac{(s_{b,n}^{\hat{c},\hat{t}})^2}{BN} - E_{\text{SPSb}}(\delta x_{\hat{c},\hat{t}})^2 \right).
 \end{aligned} \tag{12}$$

140 In the second row we considered that $\frac{1}{BN} \sum_{b',n'}^{B,N} s_{b',n'}^{\hat{c},\hat{t}}$, the operation of averaging across all sample
 141 analogues, irrespective of which bootstrap they are in, is equivalent to taking the expected value in
 142 SPSb. In the third row, because in SPSb we are calculating the variance of the means $\langle s_{b,n}^{\hat{c},\hat{t}} \rangle_n$, and each
 143 of the means is done over N samples, for the central limit theorem when $N \gg 1$ we expect a variance
 144 that is N times smaller than the population variance of the analogues sampled with probability p ,
 145 which we called $\sigma_{bn}^2(s_{b,n}^{\hat{c},\hat{t}})$. The approximation in the last row is justified by the fact that $\sigma_{bn}^2(s_{b,n}^{\hat{c},\hat{t}})$ in
 146 the third and fourth row is an unbiased estimator of the variance, and $\sum_{b,n}^{B,N} (s_{b,n}^{\hat{c},\hat{t}})^2 / (BN)$ in the last
 147 row is an unbiased estimator of the second moment of the distribution of the samples. In the limit of
 148 large B , the relation $E((z - E(z))^2) = E(z^2) - E(z)^2$ applies to unbiased estimators too.

149 Now, we know by the definition of NWKR (Section 3.5) that $E(\delta x_{\hat{c},\hat{t}}) \leftrightarrow E(Y)$ is actually
 a conditional probability $E(\delta x_{\hat{c},\hat{t}} | x_{\hat{c},\hat{t}}) \leftrightarrow E(Y|X)$, i.e. the probability of observing a certain
 displacement $\delta x_{\hat{c},\hat{t}}$ given the position on the plane $\vec{x}_{\hat{c},\hat{t}}$. Therefore we can compute the variance for
 a NWKR as:

$$\sigma^2(Y|X) = E(Y^2|X) - E(Y|X)^2 \tag{13}$$

which translates, for SPSb formalism, into:

$$\begin{aligned}
 \sigma_{\text{SPSb}}^2 &= \frac{1}{N} \sigma_{bn}^2(s_{b,n}) \\
 &\xrightarrow{B \rightarrow \infty} \frac{1}{N} \left(\sum_{c,t} p(\delta x_{c,t} | \vec{x}_{\hat{c},\hat{t}}) (\delta x_{c,t})^2 - E_{\text{NWKR}}(\delta x_{\hat{c},\hat{t}})^2 \right) \\
 &= \frac{1}{N} \left(\sum_{c,t} \mathcal{N}(\vec{x}_{c,t} - \vec{x}_{\hat{c},\hat{t}} | 0, \sigma) (\delta x_{c,t})^2 - E_{\text{NWKR}}(\delta x_{\hat{c},\hat{t}})^2 \right) \\
 &\equiv \frac{1}{N} \sigma_{\text{NWKR}}^2.
 \end{aligned} \tag{14}$$

150 We again omitted normalization terms in the third and fourth rows. This equation, combined with
 151 Eq.12, means that the standard deviation calculated with NWKR is expected to be proportional to the
 152 standard deviation calculated with SPSb multiplied by \sqrt{N} . Note that this method makes it possible
 153 to estimate any moment of the $\hat{f}(X|Y)$ distribution, not just the second.

2.1.2. Numerical convergence

We computed expectations and standard deviations for economic complexity data with both SPSb (5×10^5 bootstraps) and NWKR. The results here refer to the calculation for GDP prediction, but the same results are obtained with products predictions. It can be clearly seen from Figure 1(a) that the expectation values for SPSb converge to NWKR expectation values as the number of bootstraps increases. We show that the mean average error $\text{MAE}[E_{\text{SPSb}}(\delta x)] = \text{abs} \left[\frac{E_{\text{SPSb}}(\delta x) - E_{\text{NWKR}}(\delta x)}{E_{\text{NWKR}}(\delta x)} \right]$ converges numerically to zero (by $E_M(\delta x)$ we mean the expectation value of the displacement of x calculated with method M). The standard deviations converge as well, as can be seen from Figure 1(b). Here too we calculate $\text{MAE}[\sigma_{\text{SPSb}}(\delta x)] = \text{abs} \left[\frac{\sigma_{\text{SPSb}}(\delta x) - \sigma_{\text{NWKR}}(\delta x)}{\sigma_{\text{NWKR}}(\delta x)} \right]$. A comparison of the values obtained for expectations with the two methods is shown in Figure 3(a). The difference between predictions with the two methods is 3×10^{-5} on average with a standard deviation of 3×10^{-5} . A comparison of the standard deviations obtained with the two methods is shown in Figure 3(b). The difference between the two methods in this case is 6×10^{-4} on average with a standard deviation of 5×10^{-4} . For the purpose of GDP prediction we can therefore say that the two methods are completely interchangeable. The time complexity for SPSb is of the order $O(NB)$, while for NWKR is $O(N)$, so with $B = 1000$ bootstraps (as recommended by the literature[18]) NWKR is expected to be 1000 times faster. The same is not true for space complexity, since the original SPSb can be implemented with $O(N)$ memory requirements like NWKR.

The convergence does not reach machine precision even at 5×10^5 bootstrap cycles of SPSb because many of the analogues have extremely small probabilities to appear in a bootstrap. In Figure 2(b) we show the probabilities assigned by the kernel to all analogues of the plane for a typical prediction. In Fig.2(a) we compare, for a typical prediction, the sample frequency of each analogue with the sampling probability assigned to it by the kernel. It can be clearly seen from both figures that a sizeable proportion of the analogues has no chance to appear even in a bootstrap of 5×10^5 cycles since about 30 per cent of them have probability significantly $\leq 10^{-7}$ (each bootstrap samples $N = 10^2$ analogues). These analogues are instead included in the NWKR estimate, although with a very small weight. To obtain complete convergence one would have to sample, in total, as many analogues as the inverse of the smallest probability found among the analogues, and this number can go up to 10^{25} in typical use cases. We expect the discrepancies to decrease with the total number of samples (i.e. NB), as more and more analogues are sampled with the correct frequency. A visual representation of such discrepancies can be seen in Fig.2(a), where we plot the kernel probabilities of each available analogue $p(c, t)$ against the sampled frequencies $\phi(c, t)$ for a bootstrap of 5×10^4 samples. Discrepancies start to show, as expected, at a probability of about 10^{-6} .

2.2. HMM regularization reduces noise and increases nestedness

In analogy to what happens for countries, product classes too can be represented as points (L_t, C_t) on the Complexity-logPRODY (CL) plane. Their trajectories over time t can be then considered, and one can find the average velocity field \vec{v} by dividing the CL plane into a grid of square cells and averaging the time displacements $(\delta L_t, \delta C_t)$ of products per cell². The product model described in [20] and summarised in Section 3.3 explains the \vec{v} field in terms of competition maximization. For each product, it is possible to compute the Herfindahl index $H(p, t)$ (Eq.21 Section 3.3), which quantifies the competition on the international market for the export of product p in year t . The lower $H(p, t)$, the higher the competition. Averaging the values of $H(p, t)$

² The procedure of averaging per cell on a grid can be considered a form of nonparametric regression, but it is by no means the only technique available to treat this problem. All the following results hold independently of the regression technique used to do the spatial averages, as reported in [20]

per cell on the CL plane gives rise to a scalar field, which we call the Herfindahl field H . The inverse of the gradient of this field $-\nabla H$ explains the average velocity field (Eq.22, Section 3.3), much like a potential.

The original work where this model was proposed used a dataset of about 1000 products, classified according to the Harmonized System 2007 [20]. The Harmonized System classifies products hierarchically with a 6-digit code. The first 4 digits specify a certain class of product, and the subsequent two digits a subclass (see Section 3.7). In [20], the export flux was aggregated at the 4 digit level, and we will refer to this dataset as *noreg4*. We recently obtained the full 6-digit database, comprehensive of about 4000 products. We calculated the model on M_{cp} matrices at 6 digit level (*noreg6*), to compare it with the *noreg4* case. We also obtained the same 6-digit dataset regularized with the aforementioned HMM method [18] (see Section 3.6), which we will call *hmm6*. This method goes beyond the classical definition of the M_{cp} matrix as a threshold of the RCA matrix (Eq.16,15, in Section 3.3). Because the value of RCA fluctuates over time around the threshold, it can lead to elements of the M_{cp} matrix switching on and off repeatedly, polluting the measurements with noise. The HMM algorithm stabilizes this fluctuation. Because of this, it can significantly increase the accuracy of GDP predictions [18].

We computed the CL motion model on the three different datasets hitherto described. The results can be compared visually in Figure 4. Each of the panels in Figs.4(a,c,e) show the \vec{v} for one of the datasets, and the corresponding panels (b,d,f) plot the H field in colors, and the gradient $-\nabla H$ as arrows. The yellow line superimposed on each of the \vec{v} plots is the minimum of the vertical component of the velocity field along each column of the grid on the plane, together with error bars obtained via bootstrap. The blue line superimposed on each of the H plots is the minimum of the H field along each column of the grid together with error bars.

Noise reduction. - Panels in Figs.4(a-b) are almost identical to those in [20], since the *noreg4* data set is the same with the addition of one more year of observations (namely 2015). Figs.4(c-d) represent the velocity and Herfindahl field obtained with *noreg6*. The most noticeable change is the strong horizontal component of the velocity field: Complexity changes much faster than in *noreg4*. We believe this is due to two effects. The first one is the increased noise: when a 4-digit code is disaggregated into many 6-digit codes, there are fewer recorded export trades for each product category. This means that each individual 6-digit product category will be more sensitive to random fluctuations in time, of the kind described in Section 3.6. The second source of change is due to overly specific product classes. There are some products, such as e.g. products typical of a specific country, for which we would expect generally low Complexity. It typically happens that these products are exported by almost only one, fairly high-Fitness, country, which produces it as a speciality. When the Complexity of such products is computed with Eq.19 (Section 3.1), it will be assigned a high value, because they have few high-quality exporters. This effect increases the Complexity of the product and is stronger in more granular data. Combined with the stronger fluctuations coming from disaggregation, it contributes to noise in the Complexity measurements.

Another, stronger argument in favour of noise causing fast Complexity change over the years in *noreg6* is Figs.4(e-f). These figures show the velocity and Herfindahl field for the regularized *hmm6* data. It is clear that the horizontal components of the \vec{v} field are much smaller compared to *noreg6*, and that the only change in the data comes from the regularization, which was explicitly developed to reduce the impact of random fluctuations in export measurements. We, therefore, conclude that the HMM regularization is effective in reducing noise and generating smoother Complexity time series. Another interesting observation is that the \vec{v} obtained from *hmm6* is very similar to the *noreg4* one. Therefore we would like to conjecture that aggregating data from 6 digits to 4 has an effect

similar to that of reducing noise with the HMM algorithm. We will see in the next section that there is a further evidence to this conjecture.

Increase in nestedness. - A yet undocumented effect of HMM regularization is the increase in nestedness of the M_{cp} matrices. It can be visualized by looking at Figures 5(a,c,e). Here we show a point for each nonzero element of all M_{cp} matrices available in each dataset. To be able to resolve the differences in density, we computed a kernel estimate of the density of points on the plane. The horizontal axis is the value of rank(Complexity), while rank(Fitness) is on the vertical axis. All three datasets feature very nested matrices, as expected, but *hmm6* has one peculiarity. The top left corner of Fig.5(e) exhibits in fact a higher density than the other two. This means that regularization has the effect of activating many low-Complexity exports of high-Fitness countries. This makes sense since we expect the thresholding procedure described in Section 3.1 to be noisier in this area. Indeed, we know that the high-Complexity products are exported only by high-Fitness countries, so we expect the numerator of the RCA_{cp} (proportional to the importance of p in total world export, see Section 3.1) in this area to be small. We also know RCA_{cp} is proportional to the importance of product p relative to total exports of c , so we expect it to be high in the low-Complexity/low-Fitness area since low-Fitness countries export few products. Furthermore, it has been described in [20] that countries are observed to have similar competitive advantage in low-Complexity products regardless of their level of Fitness. So in the high-Fitness/low-Complexity area, we expect to observe a lower numerator, possibly fluctuating around the thresholding value, due to the high diversification of high-Fitness countries.

A higher density in the high-Fitness, low-Complexity area naturally results in more nested matrices. To show this, we computed the well-known NODF [11,22,23] measure of nestedness for all M_{cp} matrices in all datasets. The results can be found in Figure 6(a), and show clearly that *hmm6* matrices are much more nested than unregularized ones. Another observed result is that *noreg4* matrices are slightly but consistently more nested than the *noreg6* ones. This is further support for our conjecture that aggregating from 6 to 4 digit has an effect similar to regularizing with an HMM model. Figure 6(b) shows the significance level of the NODF measurements. In order to assess significance, we computed n_{obs} , the observed value of NODF on the M_{cp} matrices, and we compared it with n_{null} the NODF obtained from null models. The null models usually generate new adjacency matrices at random while holding some of the properties of the observed matrix (such as e.g. total number of nonzero elements) fixed. This is a way to control for the effect of the fixed property on the nestedness. Several runs of a null model generate an empirical probability distribution $p(n_{null})$. The p-value of the measurement is assessed by calculating in which quantile of $p(n_{null})$ the observed value n_{obs} falls. In Fig.6(b) we report the ratio between $n_{obs}/E_p(n_{null})$ and the scaled standard deviation of the null distribution $\sigma(n_{null})/E_p(n_{null})$, for three common null models [22]. The scaling allows to compare very different distributions on the same axis. The ratio of $\sigma(n_{null})$ to $n_{obs} - E_p(n_{null})$ is very small. Thus, the observed measurements' significance is so high that there is no need to calculate quantiles.

Model breakdown at 6 digits. - Another observation that can easily be made from Figure 4 is that, while it works well for 4-digit data, the model of product motion has trouble with reproducing the data at the 6-digit level. Regressing the \vec{v} components against the derivatives of the H field, as shown in Tab.1, seems to indicate that the 6-digit models work better³. But one key feature of the

³ However, the 4-digit BACI dataset *hmm4* has one peculiarity that needs explaining. Specifically, the bottom right corner of Fig.4(b) does not contain the maximum of H that is found in all other datasets ever observed (including the Feenstra dataset studied in [20]). This causes the gradient of H in that area to produce small values, which do not match the high vertical components of \vec{v} in the same spot, significantly lowering the R^2 coefficient of a linear regression.

model disappears when moving from 4 to 6 digits. The yellow and blue lines in Fig.4 indicate a kernel regression of respectively the minima of the \vec{v} field and the minima of the H field across each column of the grid (together with error bars obtained via bootstrap). The model predicts that \vec{v} will be almost zero where the minima of H lie, but at 6 digits this feature disappears, and the minima lines become incompatible with each other. We are currently lacking an explanation of this behaviour, that seems independent of regularisation.

Table 1. R^2 coefficients of a linear regression of \vec{v} components against the derivatives of the H field along the x-axis (Complexity) and y-axis (logPRODY).

dataset	y-axis	x-axis
4-digit non-regularized	0.103	0.023
6-digit non-regularized	0.487	0.200
6-digit regularized	0.558	0.135

2.3. Predictions on products with SPSb

Dynamics of products on the CL plane appears to be laminar everywhere, in the sense that the average velocity field seems to be smooth[20], similarly to what happens to countries on the Fitness-GDP plane [15]. If so, then it's a reasonable hypothesis that the information contained in the average velocity field can be used to predict the future positions of products on the plane. We tried to predict the future displacement of products with SPSb. Because the number of products is about 1 order of magnitude larger than the number of countries used in [18], the computational demand of the algorithm induced us to develop the proof of convergence reported in Section 2.1.

The results for the backtests on this methodology are reported in Figure 7. We predicted the Percentage Compound Annual Growth Rate (CAGR%) for each of the two metrics, and defined the error as $E = |\text{CAGR\%}_{\text{observed}} - \text{CAGR\%}_{\text{forecasted}}|$, so that if e.g. Complexity increases by 2% and we forecast 3%, $E = 1\%$. The forecasts are made at timescales $\Delta t = 3, 4, 5$ years. We used the three datasets `hmm6`, `noreg6` and `noreg4`. The predictions are not very accurate, with an error between 12% and 6% for logPRODY and in the 32-13% range for Complexity. We compared the predictions to a *random baseline*, i.e. predicting the displacement by selecting an observed displacement at random from all the available analogues. Compared to the random baseline, SPSb is always more accurate. One peculiarity about the predictions, though, is that they are generally much smaller in magnitude than the actual displacements observed. This led us to add another comparison, which we call *static baseline*, that consists in predicting zero displacement for all products. Compared to this baseline, SPSb still systematically shows some predictive power for logPRODY, especially in `noreg4`, but is definitely worse when predicting Complexity. We will clarify our explanation for this behaviour with an analogy. While the average velocity field \vec{v} exhibits laminar characteristics, in the sense that it is relatively smooth, the actual motion of the underlying products is much more disorderly. In a given neighbourhood of the CL plane, products generally move in every direction, often with large velocities, even though the average of their displacements is nonzero and small. We could tentatively describe this as a Brownian motion with a laminar drift given by \vec{v} . So trying to predict the future position of a product from their aggregate motion would be similar to trying to predict the position of a molecule in a gas. That's why the static prediction is better than a random prediction: in general, the last position of a product is a better predictor than a new random position on the plane, since the new one might be farther away. To test this Brownian motion with drift hypothesis, we added a third baseline, which we call *autocorrelation baseline*. It consists in forecasting the displacement of a product to be exactly equal to its previous observed displacement. If the hypothesis is true, we expect each product displacement to be uncorrelated with its displacement at previous time steps. For logPRODY the autocorrelation baseline is always worse than the static, which we interpret as a

signal that logPRODY changes are not autocorrelated. The reverse is true for Complexity: in fact, for noreg4 and hmm6 the autocorrelation baseline is the best predictor for Complexity change.

As already mentioned, SPSb does still have slightly but systematically more predictive power than the autocorrelation and static prediction, but only for logPRODY. We speculate that this is due to the fact that change in logPRODY is actually a signal of the underlying market structure changing, as explained in [20] and in 3.3. The fact that this advantage over the baseline is much bigger on noreg4 confirms that the logPRODY model performs significantly better on noreg4, as discussed in 3.3. On the other hand, the autocorrelation prediction (as well as the static one) can be significantly better than SPSb when predicting changes in Complexity. It is not clear whether this implies that changes in Complexity are autocorrelated in time - this effect for example disappears in noreg6, and will require an analysis with different techniques. But the fact that SPSb is always worse than the baseline, combined with the fact that regularization, which is supposed to mitigate noise, significantly reduces changes in Complexity over time raises a doubt over whether changes in Complexity are significant at all, or are drowned by noise in the M_{cp} . The fact that Complexity predictions are significantly better on the hmm6 dataset suggests confirms the contribution of noise to Complexity changes, although it is not possible to argue that regularization is strengthening the signal coming from these changes over time, since we could not characterize any signal. This might be an important finding because it could shed some light on the nature of the Complexity metric. We suggest that an alternative line of thinking should be explored, in which one treats the Complexity of a product as fixed over time. This resonates with the data structure: product classes are fixed over the timescales considered in our analyses, and new products that might be introduced in the global market during this time are not included. It also might be derived from an interpretation of the theory: Complexity is meant to be a measurement of the number of capabilities required to successfully export a product [7]. Practically, this means that there is no specific reason to believe that the Complexity of (i.e. the capabilities required for) wheat, or aeroplanes, changes over the course of the 20 years typically considered in this kind of analysis. It is possible that changes in Complexity, defined as a proxy for the number of capabilities required to be competitive in a given product, occur over longer timescales, or maybe that Complexity never changes at all. If this were true, then all observed Complexity changes would be due to noise, and it would be better to consider defining a measure of Complexity that is fixed or slowly changing in time for the model. We remark that these definition problems will probably be insurmountable as long as it is impossible to give an operational definition of capabilities, and they can only be measured indirectly through aggregate proxies, i.e. countries and products. There always is a tradeoff of interpretability to pay in order to give up normative practices in favour of operational definitions, but it affects economics and social sciences more than the physical sciences.

370 2.4. Figures, Tables and Schemes

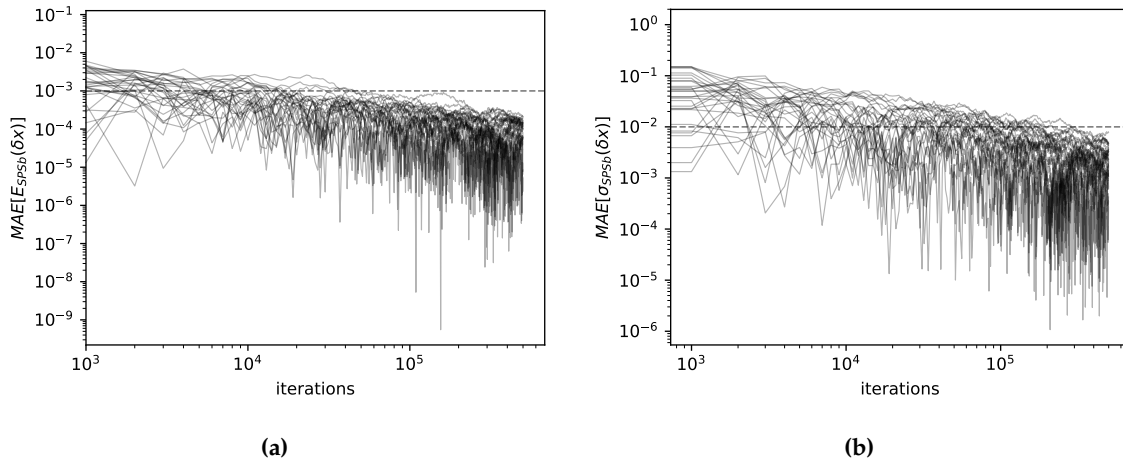


Figure 1. Panel a. (left) - For 30 predictions, we show the difference between expectation values calculated with SPSb and the same quantity computed with NWKR at different numbers of bootstraps. On the vertical axis, $MAE[E_{SPSb}(\delta x)] = \text{abs} \left[\frac{E_{SPSb}(\delta x) - E_{NWKR}(\delta x)}{E_{NWKR}(\delta x)} \right]$, i.e. the percentage mean average error done by NWKR while estimating the output of SPSb, while on the horizontal axis the number of bootstraps. After $B = 10^5$ bootstrap cycles (with the default $N = 100$ samples per cycle), the relative error is always smaller than 0.1%. This figure also allows to estimate by how much SPSb results can vary between different runs. For 10^3 bootstrap cycles, the largest deviation is around 1% of the value.

Panel b. (right) - For 30 predictions, we show the difference between standard deviations calculated with SPSb and the same quantity computed with NWKR at different numbers of bootstrap cycles. On the vertical axis $MAE[\sigma_{SPSb}(\delta x)] = \text{abs} \left[\frac{\sigma_{SPSb}(\delta x) - \sigma_{NWKR}(\delta x)}{\sigma_{NWKR}(\delta x)} \right]$, i.e. the percentage mean average error done by NWKR while estimating the standard deviation predicted by SPSb, while on the horizontal axis the number of bootstraps. After 10^5 bootstrap cycles, the relative error is always less than 1%.

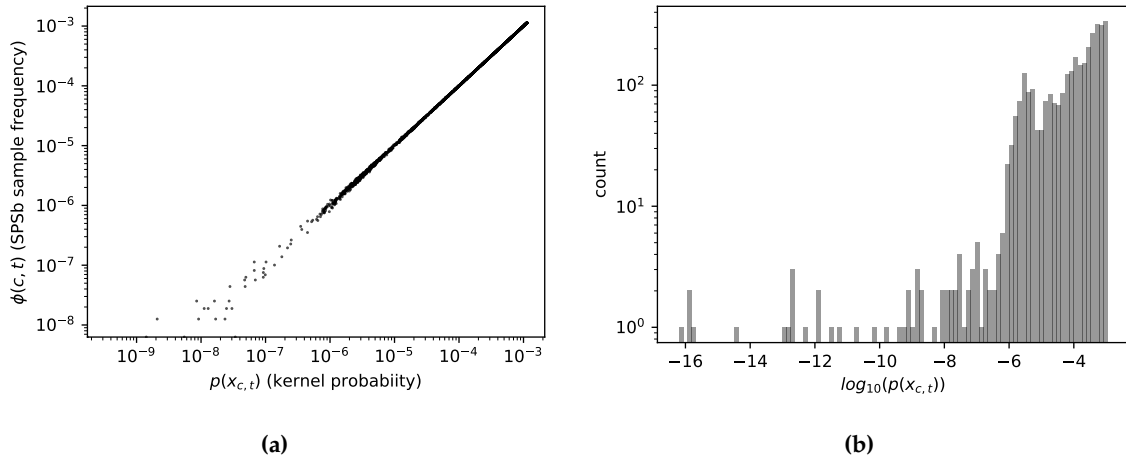


Figure 2. Panel a. (left) - Sample frequencies $\phi(c, t)$ converge to kernel probabilities $p(c, t)$, as defined in Eq.9. This plot compares them after $B = 5 \times 10^4$ bootstrap cycles of SPSb (with $N = 100$, i.e. 5×10^6 sampled analogues). The values, as expected, start to visibly diverge around 10^{-6} .

Panel b. (right) - Histogram of the probabilities assigned by the kernel to all analogues on the plane, for a typical prediction. It can be seen that a sizeable proportion of the analogues has probability e.g. $\leq 10^{-5}$. They will therefore not be included in SPSb if the number of analogues sampled is of order 10^5 .

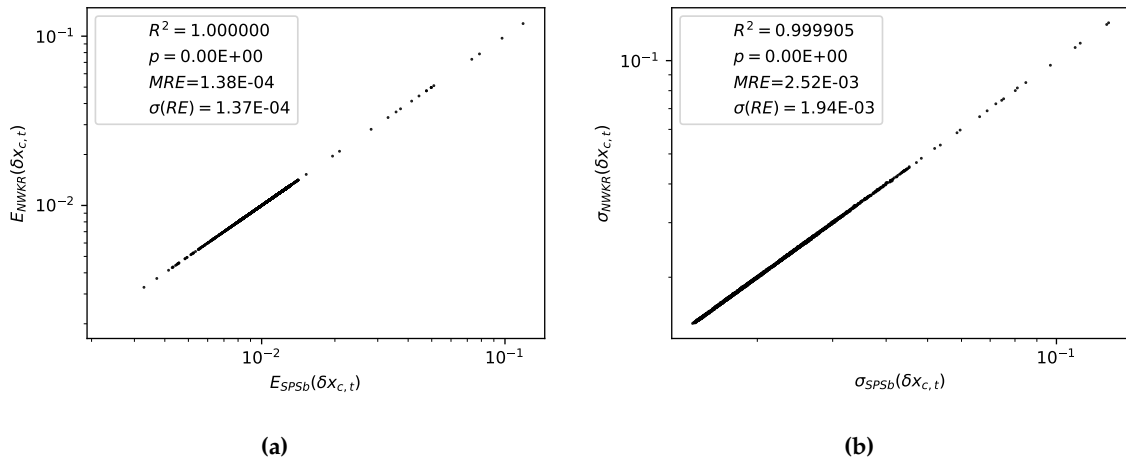


Figure 3. Panel a. (left) - For all possible predictions to be made on the plane, a comparison of the expectation values obtained with SPSb at 5×10^5 bootstrap cycles and NWKR. The match is, for all prediction purposes, perfect. In the legend, we report the value of R^2 for the observations, as well as the p-value for a linear regression (which is below machine precision, so it approximates to 0), mean relative error (the absolute value of differences normalized), and the standard deviation of the relative error.

Panel b. (right) - For all possible predictions to be made on the plane, a comparison of the standard deviations obtained with SPSb at 5×10^5 bootstraps and NWKR. The match is, again, perfect for prediction purposes.

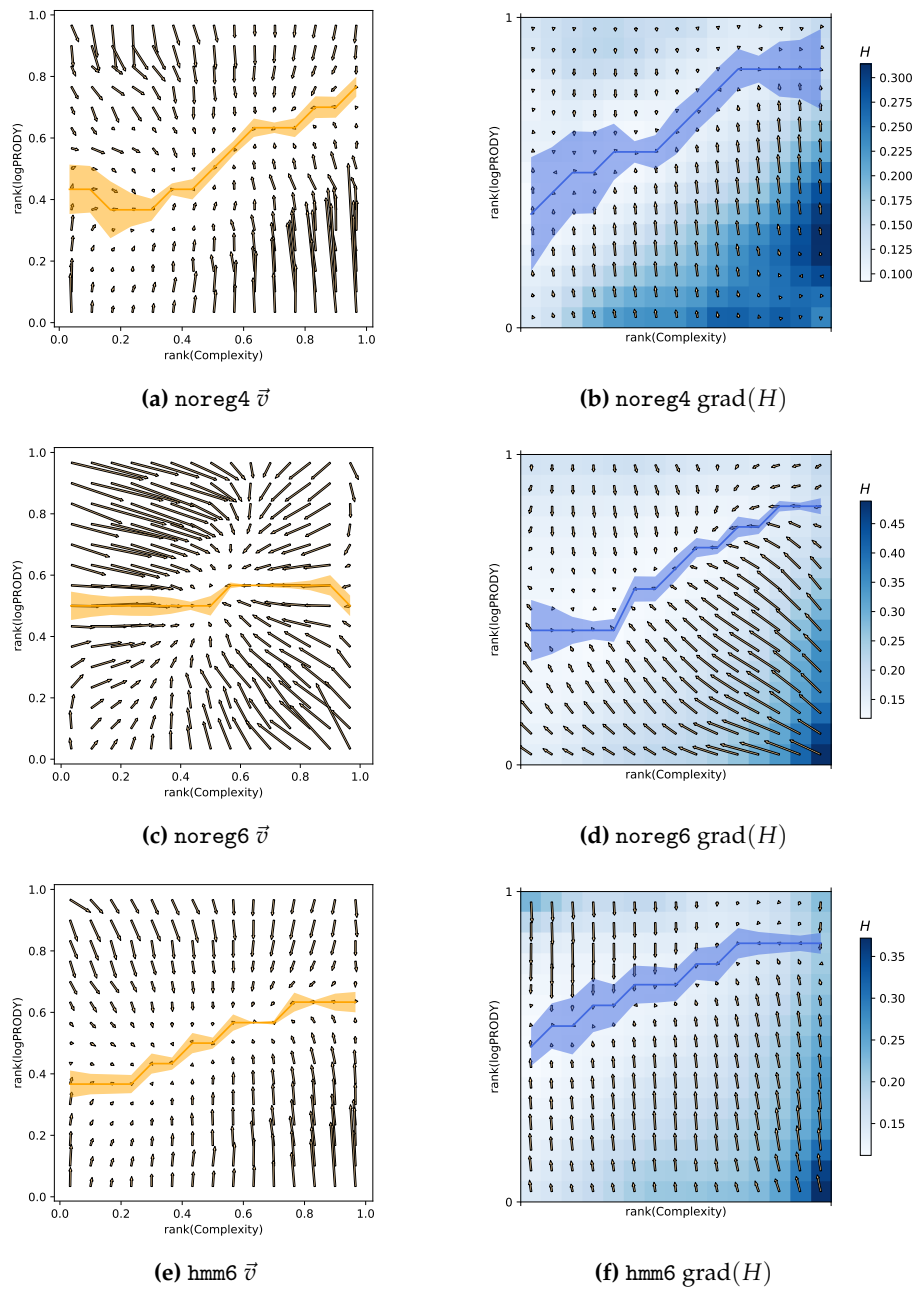


Figure 4. Comparison of the CL model of motion on the different datasets used in this work. The horizontal axes mark the Complexity, and the vertical ones logPRODY. Note that in these figures we use tied ranking as coordinates, instead of the observed values directly. Panels (a,c,e) show the \vec{v} field, together with a kernel regression of the minima of the field across the vertical direction in yellow. An uncertainty measure of this minima line has been calculated by means of a bootstrap. Panels (b,d,f) show a heat map of the H field, and its gradient. The blue line indicates the minima of the H field along the vertical direction, together with an uncertainty calculated via bootstrap.

The first feature of this Figure is the difference in the \vec{v} fields. The one calculated from noreg6 has much higher velocities on the Complexity axis, while the hmm6 velocities along the same direction are much smaller. This might be an indication that much of the change in Complexity over time is actually due to noise. The second feature is that, when going from 4 to 6 digit granularity, the observed minima lines become incompatible with those predicted by the model.

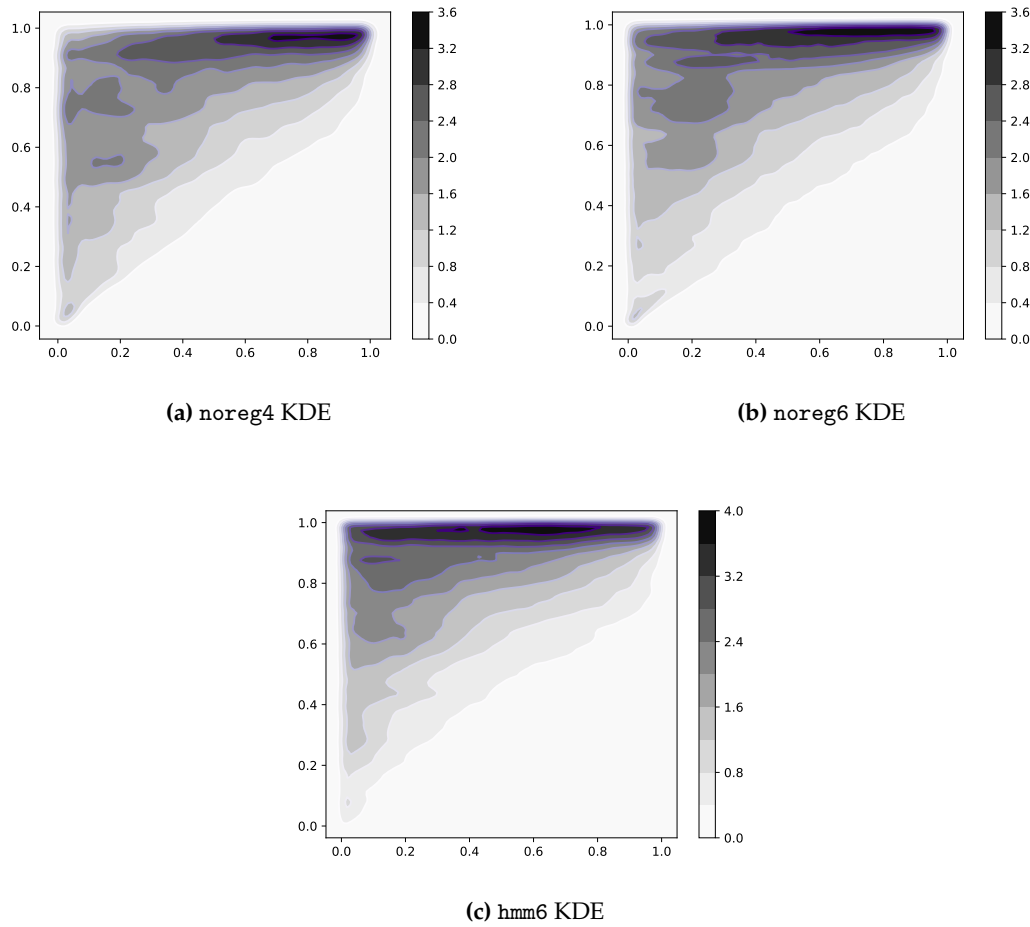


Figure 5. Comparison of M_{cp} matrices density for the 3 datasets used in this work. In each panel, we plotted one point for each nonzero element of each M_{cp} matrix in a dataset. Countries, ranked by increasing Fitness, are on the vertical axis, while products ranked by increasing Complexity on the horizontal axis. To be able to resolve the difference in the density of points, we applied a kernel density estimate (KDE). The triangular shape suggesting nestedness is clearly visible in all three cases. The differences lie in the top left corner, where low-Complexity products exported by high-Fitness countries are found. The unregularized data (noreg4, noreg6) notably have lower density here when compared with regularized matrices (hmm6). This is reflected in the increased nestedness of regularized matrices, as shown in Fig.6.

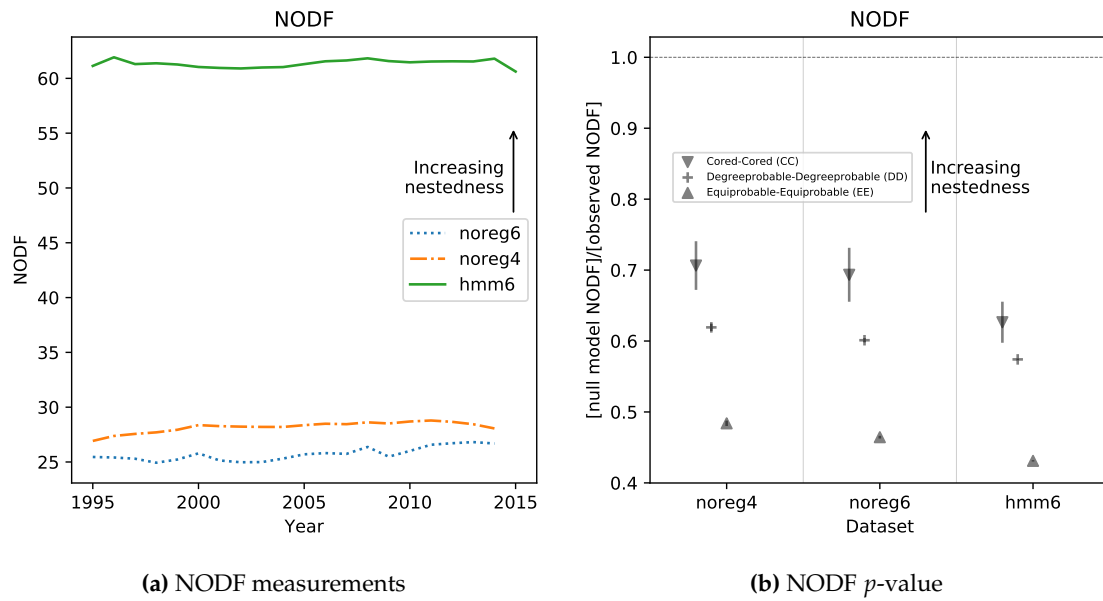


Figure 6. Panel a. (left) - Measures of nestedness for the M_{cp} matrices in the three datasets discussed in this work. We used the NODF [11] measure, which goes from 0 (no nestedness) to 100 (perfectly nested matrix). It can be clearly seen that the regularized data, `hmm6`, is much more nested than the rest, as already suggested by the observation of Fig. 5. The `noreg4` dataset, though, is significantly and consistently more nested than the `noreg6`. This suggests that aggregating from 6 to 4 digits might have a regularizing effect.

Panel b. (right) - Significativity of NODF measures. We calculate an ensemble of 100 null models for each dataset and report the ratio (null model NODF)/(observed NODF). We do this for 3 commonly used null models [22], and we report the standard deviation of the ensemble (similarly scaled) in the form of an error bar. The standard deviation of the DD and EE null models ensembles is so small that it cannot be seen in the plot. We observe that all null models have significantly smaller NODF than the observed matrices, and the results are therefore highly significant. All calculations were done with the FALCON software package [22].

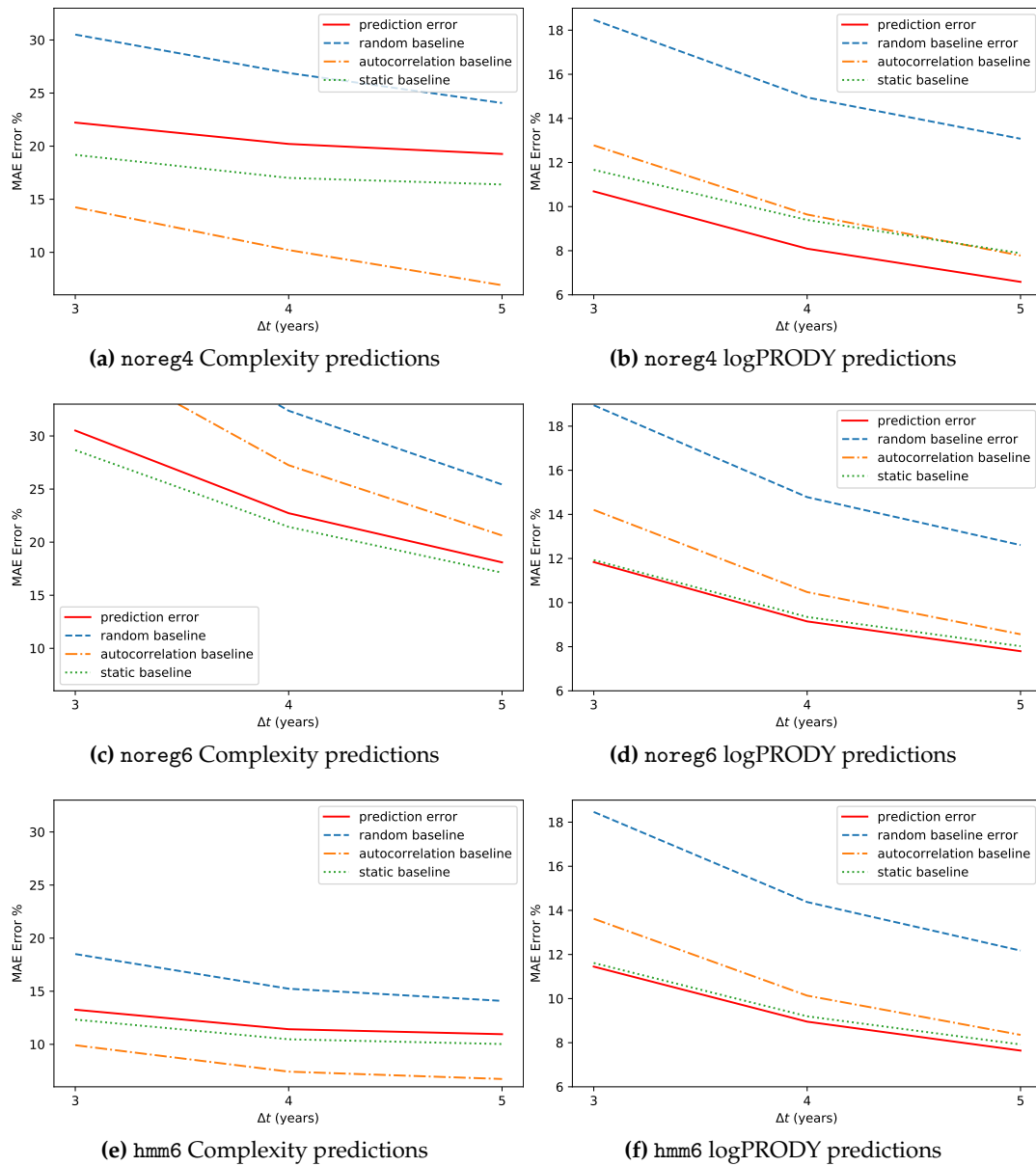


Figure 7. SPSb predictions on products. We predicted future values of logPRODY and Complexity on the log(Complexity)-logPRODY plane (*not* using ranking) with backtested SPSb at $\Delta t = 3, 4, 5$ years in the future. We used the three datasets *hmm6*, *noreg6* and *noreg4*. On the vertical axis, the Mean Average Error of the prediction (MAE). Three baselines are shown. The first one, called “random”, consists of predicting displacement by randomly selecting one available analogue. The second, called “autocorrelation”, consists of predicting the next displacement of a product to be exactly the same as the last observed one. The last, called “static” predicts 0 displacement for every product. **Panel a,c,e. (left)** - Complexity predictions are always worse than both the static baselines, and worse than the autocorrelation one in *hmm6* and *noreg4*. This might signify that observed changes in Complexity mostly caused by random noise. Very interesting is the good result of the autocorrelation baseline: this suggests that Complexity changes over time might be autocorrelated. Finally, prediction accuracy is significantly better for regularized data. It can be interpreted as a signal that, by reducing the noise, the motion becomes more predictable. **Panel b,d,f. (right)** - logPRODY predictions are significantly better than random predictions in all cases. Predictions are significantly better than all baselines for *noreg4*, and slightly but systematically better than the static prediction for the other two datasets. We interpret this as a clue that logPRODY change over time actually signals a change in market structure, as discussed in 2.3. These results also confirm that the logPRODY model performs significantly better on *noreg4*.

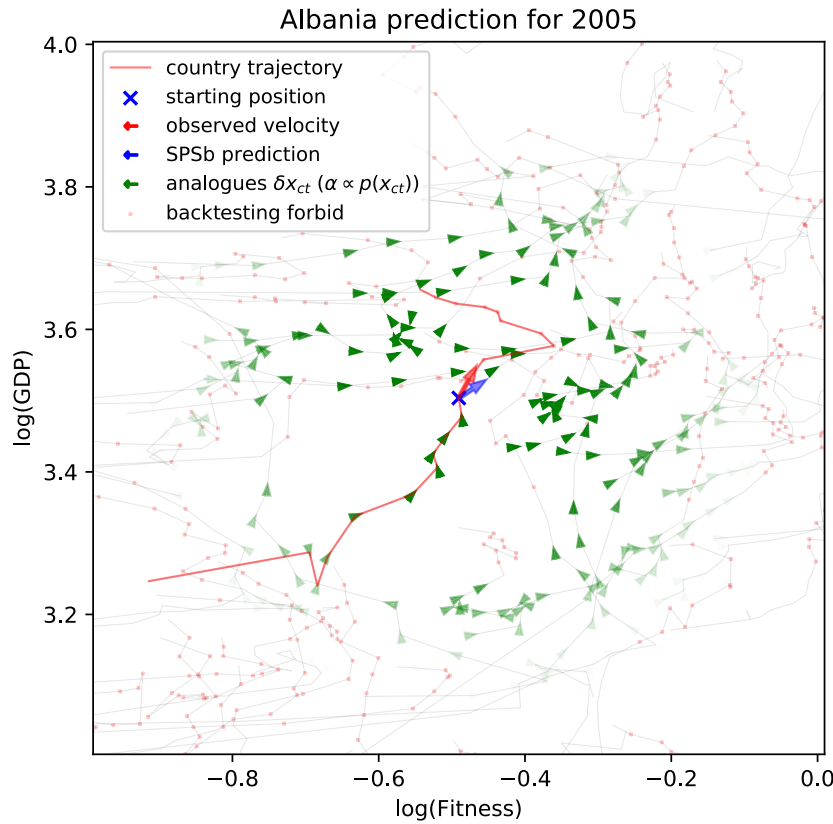


Figure 8. An example of SPSb prediction. A crop of the Fitness-GDP plane is shown; in light grey the trajectories of countries on it. In red, the trajectory of the country under examination, in this case, Albania. An x marks the position of Albania at time \hat{t} of prediction, 2005. The prediction is the average of all the available analogues, i.e. the observed trajectories of countries at times $t^{\text{past}} < \hat{t}$. The analogues are represented in green (not to scale), and the opacity is proportional to their weight in the final prediction. Analogues excluded from the calculation because are observed in the at times $t^{\text{future}} \geq \hat{t}$ are represented as red dots. A blue arrow represents the predicted displacement on the plane (for both GDP and Fitness), while a red arrow represents the observed displacement during Δt .

3. Materials and Methods

3.1. Fitness and Complexity algorithm

As discussed in Section 1, Fitness and Complexity measures are calculated from the M_{cp} . This matrix is intended to be binary, with $M_{cp} = 1$ if country c is an exporter of product p , and 0 elsewhere. To measure how significant the exports of p are for a given country, literature turns to the RCA_{cp} , where the acronym stands for *Revealed Comparative Advantage*, or Balassa index [24], and we defined the weights. If we define the value in dollars of product p exported by country c as EXM (also known as the *export matrix*), then the Balassa index is defined as:

$$RCA_{cp} = \frac{\frac{EXM_{cp}}{\sum_j EXM_{cj}}}{\frac{\sum_i EXM_{ip}}{\sum_{kl} EXM_{kl}}} \quad (15)$$

We take the ratio between the exports of p done by country c and total exports of c , and divide it by the world-average of this same ratio. Traditionally, the thresholding of this matrix returns the M_{cp} :

$$M_{cp} = \begin{cases} 1 & \text{if } RCA_{cp} \geq 1, \\ 0 & \text{otherwise.} \end{cases} \quad (16)$$

This is the definition we refer to when mentioning *unregularized* data. Because both EXM and RCA are noisy matrices, a new procedure for deriving a regularized M_{cp} has been introduced, as explained in Section 3.6.

We mention in Section 1 that the M_{cp} matrix is nested, and this observation is crucial to the definition of the Fitness-Complexity Algorithm because of two important implications. The first one is that observing a p being exported by a very diversified country c is uninformative, while if c is poorly diversified we have good reason to think that the product should be a low-Complexity one. On the other hand, if p is only exported by high-Fitness countries, chances are that it should be assigned high Complexity. The algorithm itself is a map that is iterated to convergence on the M_{cp} , and it embeds the former considerations with a non-linearity. The equations of the map are:

$$F_c^{(0)} = 1 \quad \forall c, \quad C_p^{(0)} = 1 \quad \forall p. \quad (17)$$

$$\tilde{F}_c^{(n)} = \sum_p M_{cp} C_p^{(n-1)}, \quad \tilde{C}_p^{(n)} = \frac{1}{\sum_c M_{cp} \frac{1}{F_c^{(n-1)}}} \quad (18)$$

$$F_c^{(n)} = \frac{\tilde{F}_c^{(n)}}{\langle \tilde{F}_c^{(n)} \rangle_c}, \quad C_p^{(n)} = \frac{\tilde{C}_p^{(n)}}{\langle \tilde{C}_p^{(n)} \rangle_p}. \quad (19)$$

Now Fitness of country c is defined as the plain sum of Complexities of products exported by c . Complexity of product p is instead bound by the equations to be less than the lower Fitness found among the exporters of p . Additionally, the more exporters of p , the less its Complexity. Convergence of the map can be defined numerically in various ways [25,26], and the stability of the metric with respect to noise has been studied in [27,28].

3.2. LogPRODY

LogPRODY is a modification of the PRODY index proposed by Hausmann [29], who employed it to investigate the relationship between exports and growth of a country. logPRODY is defined, for a product p , as follows:

$$L_p \equiv \sum_c \frac{RCA_{cp} \log_{10}(GDP_c)}{\sum_j RCA_{jp}} = \sum_c nRCA_{cp} \log_{10}(GDP_c), \quad (20)$$

where RCA is the Balassa index explained in Section 3.1, Eq.15. The Hausmann's PRODY is defined the same way, except that $\log_{10}(GDP_c)$ is replaced by GDP_c in the sum. We employ logarithms because the numerical distribution of GDPs spans several orders of magnitude, and a geometric average contributes to the stability of the measure [20]. Note that we defined $nRCA_{cp} = RCA_{cp} / \sum_j RCA_{jp}$, the *normalized RCA*. Comparing this quantity with the definition of RCA , we can see that normalization removes the effect of numerator from Eq.15. In other words, $nRCA_{cp}$ is proportional to the ratio between the exports of p done by country c and total exports of c . The more product p contributes to total exports of c , the more c will be weighed in \logPRODY_p . Further considerations about this measure can be found in [20].

3.3. Complexity-logPRODY motion model

Products can be represented as points on the Complexity-logPRODY (CL) plane. Their aggregate motion in time, averaged as a vector field \vec{v} can be seen in Figures 4(a,c,e). In those figures, the CL plane has been divided into a grid of cells, and we averaged the displacement vector of all products for each cell⁴. This motion can be modeled with a potential-like equation [20]. One first needs to define the Herfindahl index [30]:

$$H_p = \sum_c (s_{cp})^2; \quad s_{cp} = \frac{EXM_{cp}}{\sum_c EXM_{cp}} \quad (21)$$

where EXM_{cp} is the export matrix, defined in Section 3.1. The Herfindahl index measures the competitiveness of a market by summing the square of the market shares of each participant to the market. It ranges from 1 (for a monopoly) to $1/N$ (the case of N participants all with equal market share). When defined as in Eq.21, it refers to the total market share of countries. Averaging the Herfindahl index per cell on the CL plane produces a scalar field, H , for which one can compute the gradient with respect to the C (Complexity) and L (logPRODY) coordinates on the plane. Then the model explaining \vec{v} is:

$$\vec{v} \simeq -k_C \frac{\partial H}{\partial C} \vec{C} - k_L \frac{\partial H}{\partial L} \vec{L} \equiv -\vec{\nabla}_k H \quad (22)$$

where k_C, k_L are two scalar constants. This implies that the average velocity of products \vec{v} points towards area of lower H , i.e. higher competition on the CL plane. The lines in Figure 4 show respectively where \vec{v} is minimum and where H is minimum for each column of the grid.

The interpretation given to this model in [20] is that $\log\text{PRODY}_p$ serves as a proxy for the global market structure of product p . The full market structure is defined by the distribution of the weights of $\log\text{PRODY}_p$ across countries. As mentioned in Section 3.2, these weights are given by the nRCA_{cp} and they are proportional to the competitive advantage of country c in making product p . The market structure that maximizes H , or competition, is named *asymptotic* in [20], and it depends on Complexity. Low-Complexity products typically show an asymptotic distribution of comparative advantage that is uniform across all countries, or sometimes mildly peaked on low-Fitness countries. High-Complexity products show instead a sharp peak of comparative advantage on high-Fitness countries. The name asymptotic comes from the observation that whenever the market structure of a product is different from the asymptotic, it tends to revert to it. In doing so, it increases competition (H). LogPRODY is by definition the expectation value of the GDP on the distribution of comparative advantage, so its value tends to revert to the value it assumes on the asymptotic distribution. Interpretation for the horizontal displacements (along the Complexity axis) is, instead, less clear-cut. This difference in interpretability between logPRODY and Complexity displacements plays a role into our discussion of Section 2.3.

3.4. SPSb

As mentioned in 2.1, *Bootstrapped Selective Predictability Scheme* (SPSb) is a prediction technique allowing quantitative forecast of GDP growth for a country by averaging the growth of countries nearby on the Fitness-GDP (FG) plane [16,18]. We will describe the algorithm in detail here. Given $\vec{x}_{\hat{c},\hat{t}}$, the position of country \hat{c} in the FG plane at time \hat{t} , we want to forecast $\delta x_{\hat{c},\hat{t}}^5$, the future displacement of country \hat{c} from time \hat{t} to $\hat{t} + \Delta t$. To do so, we consider the set of observed past observations

⁴ Note that all axes in Fig.4 are labeled as $\text{rank}(\cdot)$. This is because Complexity and logPRODY can be badly behaved, and the standard treatment is to use tied ranking, instead of the observed value, when calculating this model.

⁵ Note that while the position on the FG plane is vectorial (\vec{x}), we are referring to the displacement as a scalar (δx). This is because we want to keep the formalism of the original work, which is concerned only with displacement along the GDP

414 $(\delta x_{c,t}, \vec{x}_{c,t})$ on the FG plane, which we will call *analogues*. Note that, if one wants to rigorously
 415 implement a backtesting procedure, only the analogues for which $t < \hat{t}$ are allowed. It is possible
 416 to bootstrap an empirical probability distribution for $\delta x_{\hat{c},\hat{t}}$ in two steps:

1. Sample with repetition the N available analogues with a probability distribution p given by a gaussian kernel centered in $x_{\hat{c},\hat{t}}$, i.e. the probability of sampling the analogue displacement $\delta x_{c,t}$ is:

$$p(\delta x_{c,t} | x_{c,t}) = \mathcal{N}(\vec{x}_{\hat{c},\hat{t}} - \vec{x}_{c,t} | 0, \sigma), \quad (23)$$

$$\mathcal{N}(\vec{z} | \vec{\mu}, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{(\vec{z} - \vec{\mu})^2}{2\sigma^2}\right). \quad (24)$$

417 Note that the probability of sampling depends only on the Euclidean distance between $\vec{x}_{\hat{c},\hat{t}}$ and
 418 the position of the analogue.

- 419 2. Sample $B = 1000$ bootstraps with the above procedure (bootstrap) and average the
 420 displacements per bootstrap. The global distribution of these averages is the empirical
 421 probability distribution for $\delta \vec{x}_{\hat{c},\hat{t}}$. The mean of the distribution is used as the prediction value
 422 and the standard deviation as the uncertainty on the forecast.

423 3.5. Nadaraya-Watson kernel regression

Nadaraya-Watson kernel regression was originally introduced in 1964 [31,32]. Its purpose is to estimate the conditional expectation of a variable Y relative to a variable X , which we will denote as $E(Y|X)$, in the hypothesis that the probability distributions $f(X, Y)$ and $f(X)$ exist. If one has n sampled observations $(X_1, Y_1), \dots, (X_n, Y_n)$ (where X can be multivariate), the regression model is:

$$Y_i = m(X_i) + \epsilon_i \quad (25)$$

where $m(x)$ is a (yet) unknown function and the errors satisfy these hypotheses:

$$E(\epsilon) = 0; \quad \text{Var}(\epsilon) = \sigma_\epsilon^2; \quad \text{Cov}(\epsilon_i, \epsilon_j) = 0 \quad \forall i \neq j. \quad (26)$$

One can try to approximate the probability distributions with a kernel density estimation:

$$f(X, Y) \approx \hat{f}(X, Y) = \frac{1}{n} \sum_{i=1}^n K_h(X - X_i) K_h(Y - Y_i), \quad (27)$$

$$f(X) \approx \hat{f}(X) = \frac{1}{n} \sum_{i=1}^n K_h(X - X_i). \quad (28)$$

where $K_h(x) = K(x/h)/h$ is a *kernel*, i.e. a non-negative function such that $\int K(x)dx = 1$, and $h > 0$ is called *bandwidth* and scales the kernel to provide smoothing to the regression. In this paper we

direction. Nothing forbids to forecast displacement along any arbitrary direction, though. In that case, the displacement would have to be a vector quantity.

will use only one type of kernel, the *gaussian* (also known as *radial basis function*): $K(x) = e^{-x^2}$. The conditional expected value can therefore be approximated, using Eq.27,28 as:

$$E(Y|X) = \int Y f(Y|X) dY = \int y \frac{f(X, Y)}{f(X)} dY \quad (29)$$

$$\approx \int \frac{Y \sum_{i=1}^n K_h(X - X_i) K_h(Y - Y_i)}{\sum_{i=1}^n K_h(X - X_i)} dY \quad (30)$$

$$= \frac{\sum_i K_h(X - X_i) \int Y K_h(Y - Y_i) dY}{\sum_i K_h(X - X_i)} \quad (31)$$

$$= \frac{\sum_i K_h(X - X_i) Y_i}{\sum_i K_h(X - X_i)} \equiv \hat{E}(Y|X). \quad (32)$$

$$(33)$$

Therefore we can rewrite m in Eq.25 as:

$$m_h(x) = \frac{\sum_i K(\frac{x-X_i}{h}) Y_i}{\sum_i K(\frac{x-X_i}{h})}. \quad (34)$$

3.6. HMM regularization

As explained in Section 3.1, the traditional way to calculate the M_{cp} matrix consists of calculating the RCA(Eq.15) and then thresholding it (Eq.16). This procedure introduces noise in the matrix because very often the value of RCA fluctuates around the threshold. By introducing time in the estimation of the M_{cp} it is possible to mitigate this problem. The procedure has been introduced in [18], and it consists of modelling each RCA_{cp} time series as the emission probabilities of hidden states in a Hidden Markov Model [33] (HMM). The competitive advantage of a given country c in making product p is represented as a series of 4 quantized “development stages”, obtained by calculating the quantiles of the RCA_{cp} time-series. We will call this quantized matrix RCA^q . To each of these development stages corresponds a probability to express a given value of RCA_{cp} . Countries transition between these development states with a Markov process that has transition matrix T . Both T and the parameters of the RCA distribution are estimated with the Baum-Welch algorithm [33]. Additionally, one separate model is evaluated for each country. The algorithm produces one RCA_{cp}^q matrix for each year of observation, containing the most probable development stage at each timestep. The matrices can then be binarized. It can be shown that this regularization technique reduces noise and increases the predictive performance of the SPSb algorithm [18].

3.7. Datasets and product digits

In this work, we use a dataset containing all the information of the EXM matrix (from which all the Economic Complexity metrics can be calculated). We call it BACI, and it is documented in [34]. The original data in BACI comes from UN-COMTRADE, and it has been further elaborated by CEPII, which sells the right to use it. The elaborated version of the dataset is not in the public domain, but a free version without data cleaning is available on the BACI section of the organization’s website [35]. 149 countries are included in our analysis, spanning 21 years from 1995 to 2015. Products are classified by UN-COMTRADE according to the Harmonized System 2007 [36] (HS2007). HS2007 is divided in 16 Sections, which are broad categories such as, e.g., “Vegetable Products”, “Textiles”, “Metals”, and so on. Products are then hierarchically denoted each by a set of 6-digit codes. The code is divided into three 2-digit parts, each specifying one level of the hierarchy: so the first part (Chapter) indicates the broadest categories, such as e.g. “Cereals” (10xxxx). The second two digits (Heading) specify further distinctions in each category, for example, “Rice” (1006xx). The last two digits (Subheading) are more specific, e.g. “Semi-milled or wholly milled rice, whether or not polished or glazed” (100630). For the analysis mentioned in the paper, we look at data for products aggregated at both 4-digit level (1131

products retained) and 6-digit level (4227 products). Data cleaning procedures outside of the HMM regularization mentioned above consist in the elimination of extremely small countries and countries with fragmented data; aggregation of some product categories that are closely related, and (for what we call non-regularized data) a very simple regularization of the M_{cp} matrices based the recognition and substitution of fixed handmade patterns. GDPpc data has been downloaded from the World Bank Open Data website [37].

4. Conclusions

In this work, we focused on the analysis of Product Complexity, which had received little attention since [20]. The application of the motion model to the 6-digit data set with and without HMM regularisation seems to indicate that much of the change in Complexity over time is due to noise. Further analysis will be certainly needed on this topic, as it could lead to a better understanding of the Complexity measure as discussed in 2.3. We suggest that these results should be strengthened and confirmed in future work by an evaluation of the quantity of noise might be carried out, in the fashion of [27,38]. Insights gathered this way might be used to calibrate a model that evaluates the effect of noise on Complexity change over time. Also very interesting is the finding that changes in Complexity might be autocorrelated over time. Further analysis is needed to clarify whether this is true, and if appropriate to understand the causes of the autocorrelation.

Applying SPSb to the CL plane seems to confirm the findings of [20] regarding the meaning of logPRODY and gives further grounds to argue that changes in Complexity over time are not relevant. The same suggestions as before apply: further validation with a study of the noise is probably a good research path. We analysed the change in nestedness caused by the HMM regularisation technique on the M_{cp} matrices, and thoroughly validated the statistical significance of the difference with several null models. We suggest that aggregating data from 6 to 4-digit level might have a regularising effect. Finally, in order to be able to apply SPSb to a data set larger by one order of magnitude than what was previously done, we developed proof that SPSb itself converges, for a high number of iterations, to a well-known statistical learning technique, NWKR. The two techniques can be used interchangeably. NWKR has the advantage of being significantly faster, and of producing a deterministic result. The proof also has the benefit of further clarifying the nature of SPSb. This technique belongs to the same family of algorithms that predict by similarity based on distance, such as NWKR and k-nearest neighbours. We suggest that regression trees might do well in its place, too. We also suggest that a further technical development in this field might be the introduction of one of the many flavours of variable-bandwidth NWKR techniques because of the significant changes in density of analogues over the considered data sets.

Supplementary Materials: We released custom code for NWKR calculations [39] and a wrapper that allows using FALCON[22] in Python 3.6[23]. Data availability is discussed in Section 3.7.

Acknowledgments: We would like to thank Professor Luciano Pietronero, Dr Andrea Gabrielli, Dr Andrea Zaccaria, Dr Andrea Tacchella, Dr Dario Mazzilli, for enlightening conversations and for freely exchanging their findings and data with us. Thanks to Anshul Verma for the many useful discussions. We also wish to thank the ESRC Network Plus project “Rebuilding macroeconomics”.

Author Contributions: All authors contributed equally.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

GDP: Gross Domestic Product

SPSb: Bootstrapped Selective Predictability Scheme

HMM: Hidden Markov Model

NWKR: Nadaraya-Watson kernel regression

L: logPRODY

C: Complexity

M_{cp} : export bipartite network adjacency matrix

FG: Fitness-GDP

CL: Complexity-logPRODY

RCA: Revealed Comparative Advantage

nRCA: Normalized RCA

EXM: EXport Matrix

Appendix A. Country predictions via the products

Even though a definitive interpretation for both Complexity and logPRODY is lacking, if predictions on the trajectories of products are better than random, one can try and use them to make predictions on the countries' trajectories. By definition, a country's Fitness is equal to the sum of the complexities of its exports (see Section 3.1), i.e.

$$F_c = \sum_p M_{cp} Q_p, \quad (35)$$

while countries' GDP's are connected to the logPRODYs via

$$\log PRODY_p = \sum_c \frac{RCA_{cp} \log_{10}(GDP_c)}{\sum_j RCA_{jp}} \equiv \sum_c nRCA_{cp} \log_{10}(GDP_c), \quad (36)$$

where we defined $nRCA_{cp} \equiv RCA_{cp} / \sum_j RCA_{jp}$. Therefore, if we can find $nRCA^{-1}$ such that $nRCA^{-1} nRCA = 1$, we can invert the relation and obtain:

$$\log_{10}(Y_c) = \sum_p nRCA_{pc}^{-1} \log PRODY_p. \quad (37)$$

We can then feed our estimates of future positions of products to these equations, to obtain an estimate on future positions of countries on the FG plane. Because of the lack of predictive power described in Section 2.3, country predictions are worse than all baselines (result not shown in this work). Furthermore, it is known in general from the statistical learning literature[21], and in particular for Economic complexity[16] that averaging the prediction of two different models can improve significantly the error of a regression. Averaging our countries' predictions with the predictions made by SPSb on the FG plane results in worse performance, thus we argue that the product's predictions are tainted by large amounts of noise. This noise comes primarily from the locally disorderly motion in the CL plane, but there is another important source of noise too. An important contribution to the change in Fitness is due to new products being exported (or lost) over time. But in a backtesting, the M_{cp} and $nRCA$ matrices fed to Equation 35 contain only information about products exported at the initial time. This is true for the GDP too, if one substitutes the M_{cp} matrix with the $nRCA$ in Eq.36.

Bibliography

1. Tacchella, A.; Cristelli, M.; Caldarelli, G.; Gabrielli, A.; Pietronero, L. A New Metrics for Countries' Fitness and Products' Complexity. *Scientific Reports* **2012**, *2*.
2. Dosi, G. Sources, Procedures, and Microeconomic Effects of Innovation. *Journal of economic literature* **1988**, *26*, 1120–1171.
3. Lall, S. Technological capabilities and industrialization. *World Development* **1992**, *20*, 165–186.
4. Teece, D.J.; Rumelt, R.; Dosi, G.; Winter, S. Understanding corporate coherence. Theory and evidence. *Journal of Economic Behavior and Organization* **1994**, *23*, 1–30, [arXiv:10.1016/0167-2681(94)90094-9].
5. Hidalgo, C.A.; Klinger, B.; Barabasi, A.L.; Hausmann, R. The Product Space Conditions the Development of Nations. *Science* **2007**, *317*, 482–487, [0708.2090].

6. Hidalgo, C.A.; Hausmann, R. The building blocks of economic complexity. *Proceedings of the National Academy of Sciences* **2009**, *106*, 10570–10575, [0909.3890].
7. Cristelli, M.; Gabrielli, A.; Tacchella, A.; Caldarelli, G.; Pietronero, L. Measuring the Intangibles: A Metrics for the Economic Complexity of Countries and Products. *PLoS ONE* **2013**, *8*.
8. Liao, H.; Vidmer, A. A Comparative Analysis of the Predictive Abilities of Economic Complexity Metrics Using International Trade Network. *Complexity* **2018**, *2018*, 1–12.
9. Bascompte, J.; Jordano, P.; Melián, C.J.; Olesen, J.M. The nested assembly of plant-animal mutualistic networks. *Proceedings of the National Academy of Sciences of the United States of America* **2003**, *100*, 9383–9387.
10. Domínguez-García, V.; Muñoz, M.A. Ranking species in mutualistic networks. *Scientific Reports* **2015**, *5*, 8182.
11. Almeida-Neto, M.; Guimarães, P.; Guimarães, P.R.; Loyola, R.D.; Ulrich, W. A consistent metric for nestedness analysis in ecological systems: reconciling concept and measurement. *Oikos* **2008**, *117*, 1227–1239.
12. Zaccaria, A.; Cristelli, M.; Tacchella, A.; Pietronero, L. How the Taxonomy of Products Drives the Economic Development of Countries. *PLoS ONE* **2014**, *9*, e113770.
13. Loreto, V.; Servedio, V.D.P.; Strogatz, S.H.; Tria, F. Dynamics on expanding spaces: modeling the emergence of novelties. *ArXiv e-prints* **2017**, [1701.00994].
14. Pugliese, E.; Chiarotti, G.L.; Zaccaria, A.; Pietronero, L. Complex Economies Have a Lateral Escape from the Poverty Trap. *PLOS ONE* **2017**, *12*, e0168540.
15. Cristelli, M.; Tacchella, A.; Pietronero, L. The Heterogeneous Dynamics of Economic Complexity. *PLOS ONE* **2015**, *10*, e0117174.
16. Cristelli, M.; Tacchella, A.; Cader, M.; Roster, K.; Pietronero, L. *On the predictability of growth*; The World Bank, 2017.
17. Lorenz, E.N. Atmospheric Predictability as Revealed by Naturally Occurring Analogues. *Journal of the Atmospheric Sciences* **1969**, *26*, 636–646.
18. Tacchella, A.; Mazzilli, D.; Pietronero, L. A dynamical systems approach to gross domestic product forecasting. *Nature Physics* **2018**, p. 1.
19. Sbardella, A.; Pugliese, E.; Pietronero, L. Economic development and wage inequality: A complex system analysis. *PLOS ONE* **2017**, *12*, e0182774.
20. Angelini, O.; Cristelli, M.; Zaccaria, A.; Pietronero, L. The complex dynamics of products and its asymptotic properties. *PLOS ONE* **2017**, *12*, e0177360.
21. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning*; Springer Series in Statistics, Springer New York Inc.: New York, NY, USA, 2001.
22. Beckett, S.J.; Boulton, C.A.; Williams, H.T.P. FALCON: a software package for analysis of nestedness in bipartite networks. *F1000Research* **2014**, *3*, 185.
23. Angelini, O. ganileni/pyfalcon: First release, 2018. <https://doi.org/10.5281/zenodo.1451790>.
24. Balassa, B. Trade Liberalisation and "Revealed" Comparative Advantage. *The Manchester School* **1965**, *33*, 99–123.
25. Pugliese, E.; Zaccaria, A.; Pietronero, L. On the convergence of the Fitness-Complexity algorithm. *The European Physical Journal Special Topics* **2016**, *225*, 1893–1911.
26. Wu, R.J.; Shi, G.Y.; Zhang, Y.C.; Mariani, M.S. The mathematics of non-linear metrics for nested networks. *Physica A: Statistical Mechanics and its Applications* **2016**, *460*, 254–269.
27. Battiston, F.; Cristelli, M.; Tacchella, A.; Pietronero, L. How metrics for economic complexity are affected by noise. *Complexity Economics* **2014**, *3*, 1–22.
28. Mariani, M.S.; Vidmer, A.; Medo, M.; Zhang, Y.C. Measuring economic complexity of countries and products: which metric to use? *The European Physical Journal B* **2015**, *88*, 293.
29. Hausmann, R.; Hwang, J.; Rodrik, D. What you export matters. *Journal of Economic Growth* **2007**, *12*, 1–25.
30. Kelly, W.A. A Generalized Interpretation of the Herfindahl Index. *Southern Economic Journal* **1981**, *48*, 50.
31. Nadaraya, E.A. On Estimating Regression. *Theory of Probability & Its Applications* **1964**, *9*, 141–142.
32. Watson, G.S. Smooth Regression Analysis. *Sankhyā: The Indian Journal of Statistics, Series A (1961-2002)* **1964**, *26*, 359–372.
33. Rabiner, L.R.; Juang, B.H. An introduction to hidden Markov models. *ieee assp magazine* **1986**, *3*, 4–16.

34. Gaulier, G.; Zignago, S. BACI: International Trade Database at the Product-Level (the 1994-2007 Version). *SSRN Electronic Journal* **2010**.
35. CEPII BACI dataset. <http://www.cepii.fr/CEPII/en/welcome.asp>.
36. World Customs Organization. <http://www.wcoomd.org/>.
37. The World Bank Open Data. <http://data.worldbank.org/>.
38. Tumminello, M.; Miccichè, S.; Lillo, F.; Piilo, J.; Mantegna, R.N. Statistically validated networks in bipartite complex systems. *PLoS ONE* **2011**, *6*, e17994, [1008.1414].
39. Angelini, O. ganileni/ectools: First release, 2018. <https://doi.org/10.5281/zenodo.1451786>.

© 2018 by the authors. Submitted to *Entropy* for possible open access publication under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>)